EOSC-hub receives funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No. 777536.

Python-based Multidimensional Climate Model Data Analysis in ECAS

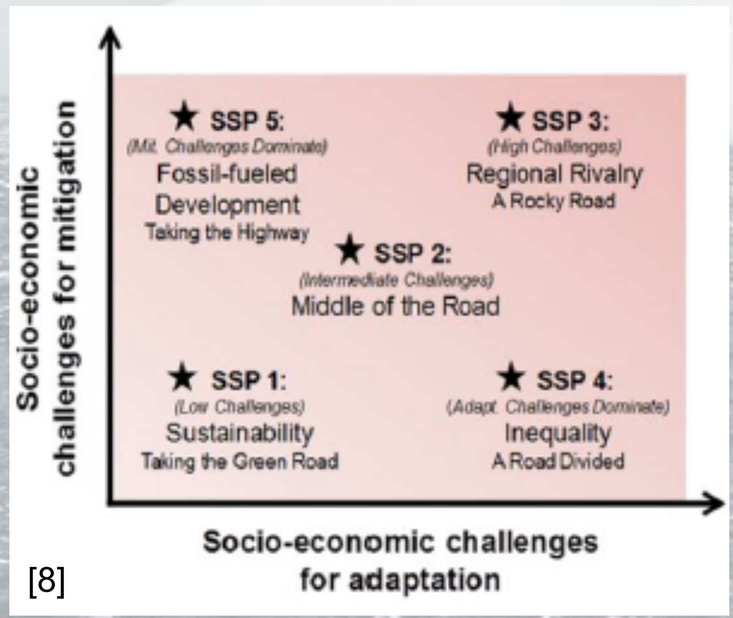
R. Kwee *(DKRZ), S. Kindermann (DKRZ), K.Peters (DKRZ), H. Thiemann (DKRZ), S. Fiore (CMCC), D. Elia (CMCC)

DKRZ DEUTSCHES KLIMARECHENZENTRUM

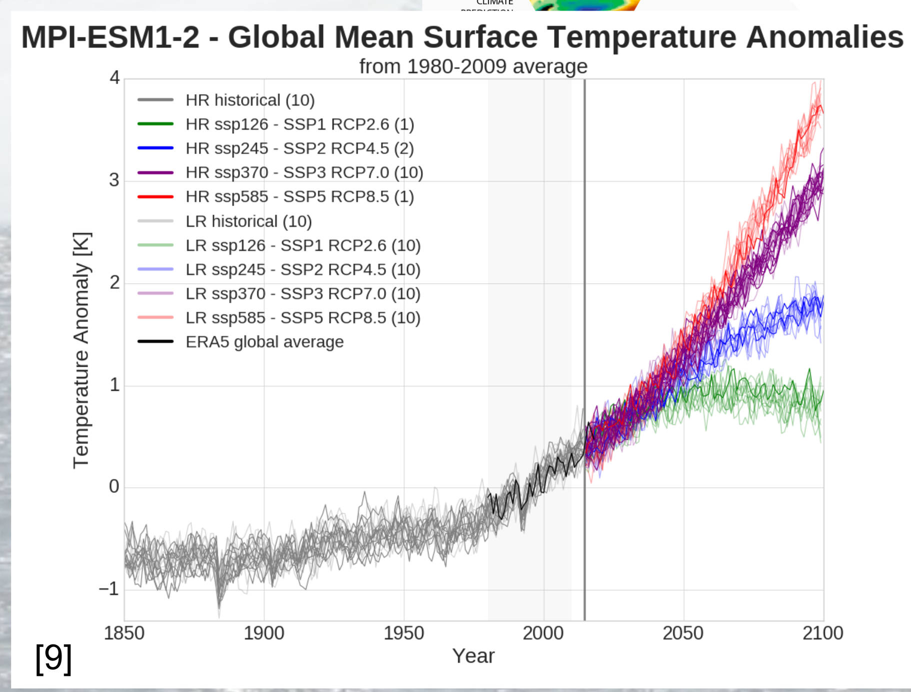
The ENES Climate Analytics Service (ECAS) allows end-users to perform server-side processing on climate data.

- Earth system data are massive. Server-side climate data analysis reduces efforts retrieving and storing the data.
- We offer a JupyterHub (at jupyterhub.dkrz.de) with access to the full CMIP datapool.
- Two simulation models (out of 21 CMIP6 + CORDEX) comprising all scenario data and additional high resolution output data.

World Climate Research Programme Grand CHALLENGES



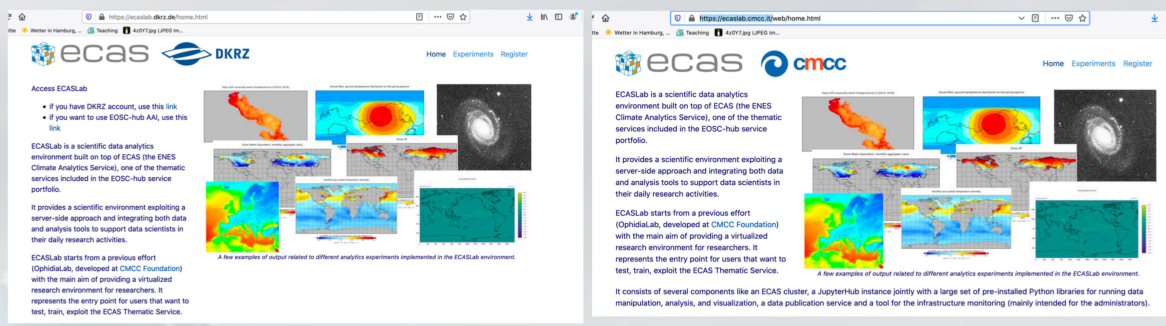
[8]



[9]

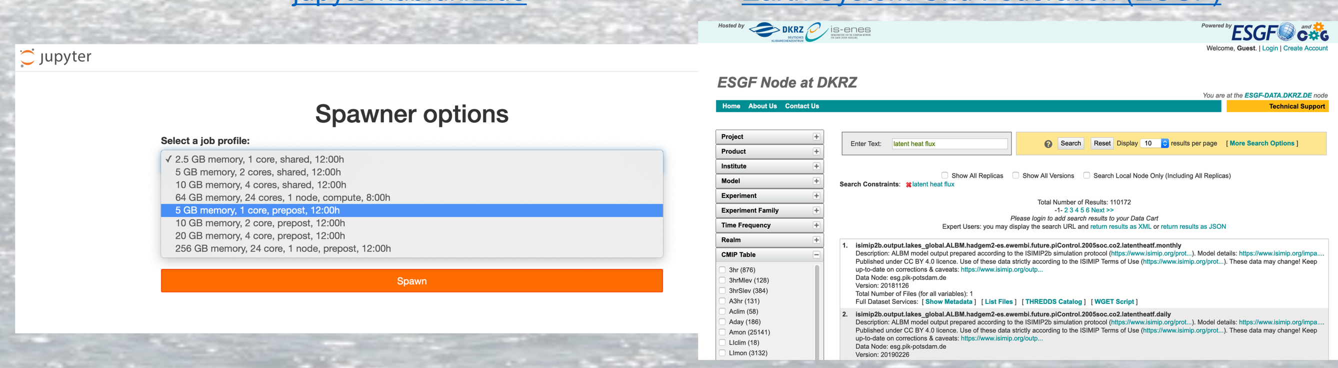
EOSC-hub

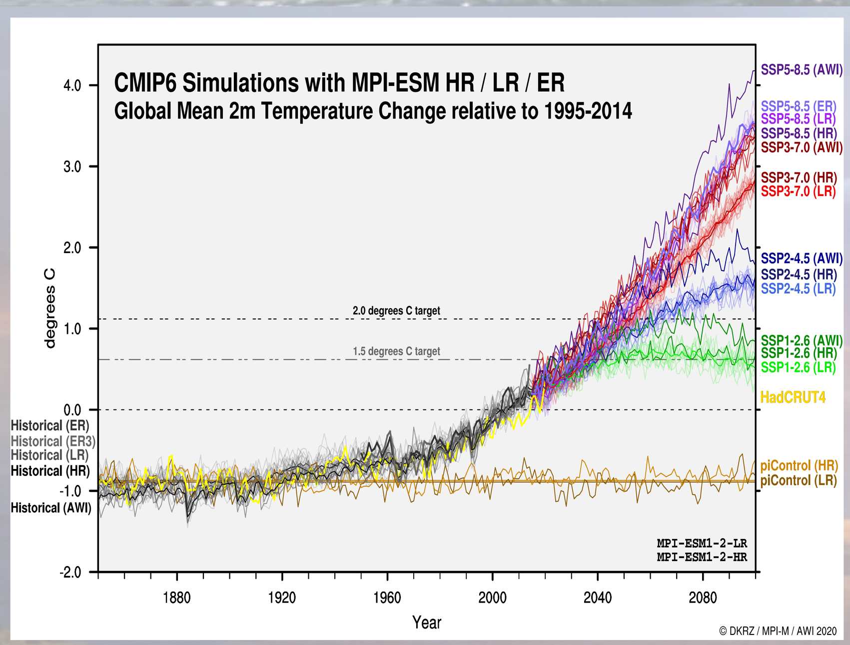
https://ecaslab.dkrz.de/ https://ecaslab.cmcc.it/



- ECAS is a service offered by DKRZ and CMCC that can handle large amounts of climate data.
- DKRZ capabilities on tape 150 PB, on disk 60 PB (of 500 PB).

jupyterhub.dkrz.de https://esgf-data.dkrz.de Earth System Grid Federation (ESGF)





Top: comparison of the different datasets of the variable TAS. From right piControl (brownish), HadCrut4 (from measurements, yellow), SSP126 (green), SSP245 (blue), SSP370 (red), SSP585 (purple). From left: historical datasets (grey). [11]

An example directory structure of the CMIP6 on mistral:

<mip_era>/
<activity_id>/
<institution_id>/
<source_id>/
<experiment_id>/
<member_id>/
<table_id>/
<variable_id>/
<grid_label>/
<version>

python™ INTAKE

an example from [2]:

	activity_id	institution_id	source_id	experiment_id	member_id	table_id	variable_id	grid_label	zstore	dcpp_init_year
0	AerChemMIP	BCC	BCC-ESM1	ssp370	r1i1p1f1	Amon	pr	gn	gs://cmip6/AerChemMIP/BCC/BCC-ESM1/ssp370/r1i1...	NaN

an example from mistral:

	activity_id	institution_id	source_id	experiment_id	member_id	table_id	variable_id	grid_label	dcpp_init_year	version	time_range	path
0	AerChemMIP	HAMMOZ-Consortium	MPI-ESM-1-2-HAM	ssp370-lowNTCF	r1i1p1f1	Lmon	npp	gn	NaN	v20190627	203501-205412	/mnt/lustre02/work/ik1017/CMIP6/data/CMIP6/Aer...

Introduction : Key features of xarray

- is an open source software, freely available to everyone,
- good for multi-dimensional handling of data and
- builds on top of pandas, a data processing python library.

xarray

Basic operations: opening, inspecting, selecting data.

- opening: work locally with example files (proof-of-principle) or use intake or (for massive data analysis) or do both!
- inspecting: explore your data! How can I access the variables? What do I need to plot? How to make a plot that a collegue can understand?
- selecting: How can I efficiently handle my data, ie. how do I use the concepts of concatination, the groupby-mechansim and data aggregation?

Hierarchical indexing

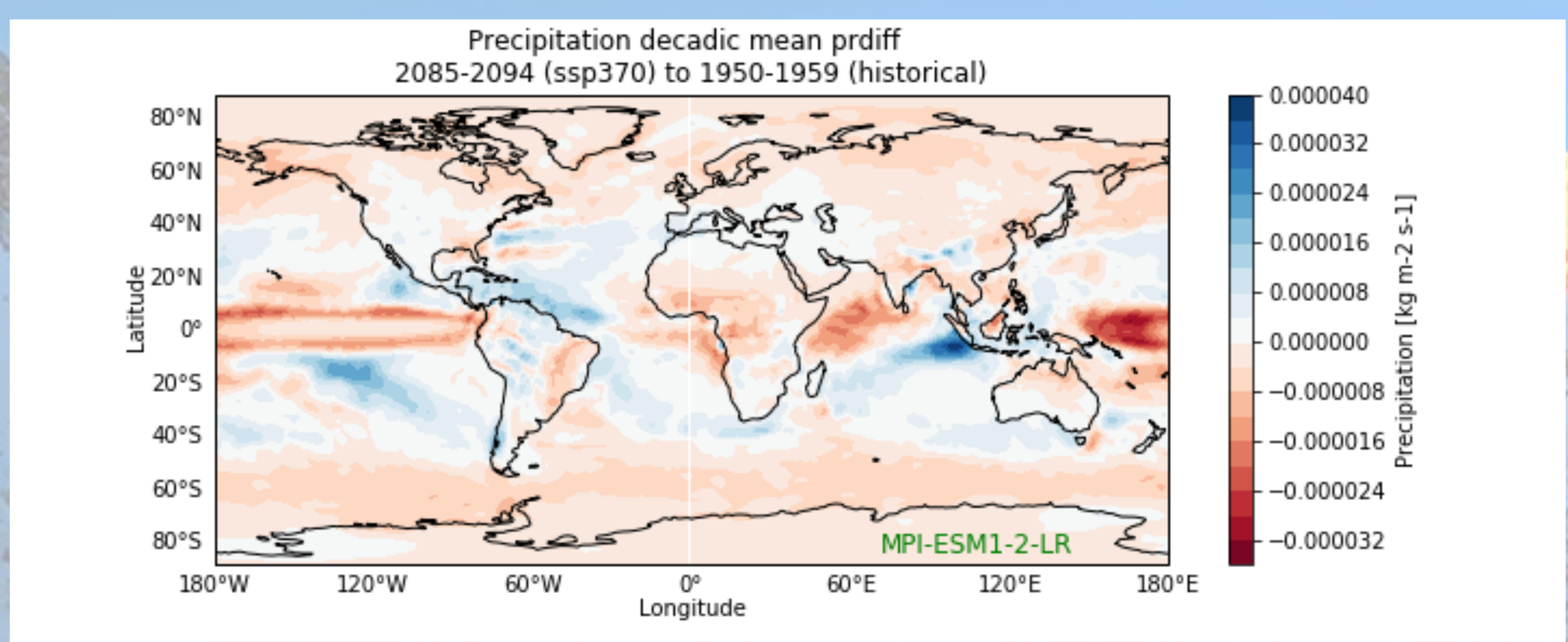
- is an important feature of pandas which allows you to work on higher dimensional data in a lower dimensional form.

jupyter

```
# plot

# chose one of the ds
lons = ds_ssp['lon']
lats = ds_ssp['lat'] #x


fig = plt.figure(figsize=(10, 4))
ax = plt.axes(projection=ccrs.PlateCarree())
ax.coastlines()

# so mittelt man ueber die ersten 10 Jahre
pr10=ds_his['pr'].isel(time=slice(0,10*12)).mean('time')

# so berechnet man den MW der 10 Jahre rueckwaerts von 240 Zeitschritten
pr10end=ds_ssp['pr'].isel(time=slice(240-10*12,240)).mean('time')

# difference
prdiff = pr10 - pr10end

# argumente: countourf(x, y, z, zbins,
p = plt.contourf(lons, lats, prdiff, 20,
           # transform of the projection, color map = red/blue scale for color blinds
              transform=ccrs.PlateCarree(), cmap='RdBu')


# -- label
ax.set_xlabel('Longitude')
ax.set_ylabel('Lattitude')

# -- per Hand
ax.set_xticks([-180,-120,-60,0,60,120,180])
ax.set_yticks([-90,-60,-30,0,30,60,90])
```

```
# draw colorbar
c = plt.colorbar(p)

#print(ds_his['pr'].attrs['units'])

# for plot title
plottitle = ds_his['pr'].attrs['long_name']
c.ax.set_ylabel(plottitle + ' ['+ ds_his['pr'].attrs['units'] +']')

# string datetime
strdt =' \n2085-2094 (ssp370) to 1950-1959 (historical)'
pname = plottitle + ' decadisches Mittel prdiff '+ strdt

plt.title(pname)

# save before show()
pname='/Users/reginakwee/c.png'
plt.savefig(pname)
plt.show()
```

The tutorial can be found at https://github.com/regkwee/dm/blob/master/xarray.ipynb



Outlook and conclusions

- Use this tutorial as template if you want to make such a plot.
- Include Dask [5] for more and faster computing!
- Stay tuned for more pythonic fun!

References:

[1] Wes McKinney, Python for Data Analysis: Data Wrangling with Pandas, NumPy, and IPython 2nd Edition, 2018

[2] last visited 02.May 2020, https://intake-esm.readthedocs.io

[3] xarray.pydata.org: visited on 02.May 2020 http://xarray.pydata.org/en/stable/

[4] egu2017 xarray tutorial, last visited on 02.May 2020: http://pure.iiasa.ac.at/id/eprint/14952/1/xarray-tutorial-egu2017-answers.pdf

[5] dask.org

[6] Jupyter notebook by A.Spring, last viewed on 02.May 2020 https://nbviewer.jupyter.org/github/aaronspring/xarray_dask_talk_unihh/blob/master/notebook/xarray_dask.ipynb

[7] ESGF Datapool: esgf-data.dkrz.de

[8] Riahi et al., The Shared Socioeconomic Pathways and their energy, land use, and greenhouse gas emissions implications: An overview, Global Environmental Change 42 (2017) 153–168, http://dx.doi.org/10.1016/j.gloenvcha.2016.05.009

[9] Martin Schupfner, private communication, Dec. 2019

[10] Eyring, V., Bony, S., Meehl, G. A., Senior, C. A., Stevens, B., Stouffer, R. J., and Taylor, K. E.: Overview of the Coupled Model Intercomparison Project Phase 6 (CMIP6) experimental design and organization, Geosci. Model Dev., 9, 1937–1958, https://doi.org/10.5194/gmd-9-1937-2016, 2016.

[11] Karin Meier-Fleischer (DKRZ), 12.05.2020,

*contact details: kwee@dkrz.de